Routes API

Method	**Endpoint**	**Body**	**Response**	**Protégé**	**Code**	**Description**
"POST"	"/api/auth/register"	{ "username": "string", "email": "string", "password": "string" }	{ "status": "success", "message": "User created" }	❌	**201**	S’inscrire
“POST”	"/api/auth/login"	{ "email": "string", "password": "string" }	{ "status": "success", "token": "jwt_token" }	❌	**200**	Se connecter et obtenir un jeton JWT
"GET"	"/api/exercices/ :id "		{ "id": "int", "name": "string", "description": "string", "muscleGroups": ["string"] }	✅	**200**	Récupérer un exercice par son id
"GET"	"/api/exercices "		{ "status": "success", "data": [{ "id": "string", "name": "string", "created_at": "date", "updated_at": "date" }, ...] }	✅	**200**	Récupérer tous les exercices
“POST”	"/api/exercices"	{ "name": "string", "description": "string", "muscleGroups": ["string"] }	{ "status": "success", "message": "Exercise created" }	✅	**201**	Créer un nouvel exercice

“PUT”	"/api/exercices/ :id"	{ "name": "string", "description": "string", "muscleGroups": ["string"] }	{ "status": "success", "message": "Exercise updated" }	✅	**200**	Mettre à jour un exercice par son id
“DELETE”	"/api/exercices/ :id"		{ "status": "success", "message": "Exercise deleted" }	✅	**204**	Supprimer un exercice par son id
"GET"	"/api/posts"		[{ "id": "int", "title": "string", "content": "string", "authorId": "int", "createdAt": "date" }]	✅	**200**	Récupérer toutes les publications
"POST"	"/api/posts"	{ "title": "string", "content": "string", "authorId": "int" }	{ "status": "success", "message": "Post created" }	✅	**201**	Créer une nouvelle publication
"PUT"	"/api/posts/ :id"	{ "title": "string", "content": "string" }	{ "status": "success", "message": "Post updated" }	✅	**200**	Mettre à jour une publication par son id
"DELETE"	"/api/posts/ :id"		{ "status": "success", "message": "Post deleted" }	✅	**204**	Supprimer une publication par son id
“GET”	“/api/posts/:id		{ "id": "int", "title":	✅	**200**	Récupérer une

			"string", "content": "string", "authorId": "int", "createdAt": "date" }			publication par son id
“GET”	“/api/posts/users/ :id		[{ "id": "int", "title": "string", "content": "string", "authorId": "int", "createdAt": "date" }]	✅	**200**	Récupérer toutes les publications faites par un utilisateur par l’id de l’utilisateur
"GET"	"/api/posts/ :id/accueil"		[{ "id": "int", "title": "string", "content": "string", "authorId": "int", "createdAt": "date" }]	✅	**200**	Récupérer toutes les publications du fil d’actualité d’un utilisateur par son id (cela inclut ses publications, celles de ses abonnements , les partages)
"POST"	"/api/posts/ :id/likes"		{ "status": "success", "message": "Like added" }	✅	**201**	Ajouter un like à une publication
"DELETE"	"/api/posts/ :id/likes"		{ "status": "success", "message": "Like removed" }	✅	**200**	Supprimer un like d’une publication
"GET"	"/api/posts/ :id/likes"		[{ "userId": "int",	✅	**200**	Récupérer les utilisateurs

			"username": "string" }]			ayant aimé une publication
"POST"	"/api/posts/ :id/comments "	{ "content": "string", "authorId": "int" }	{ "status": "success", "message": "Comment added" }	✅	**201**	Ajouter un commentaire à une publication
"GET"	"/api/posts/ :id/comments "		[{ "id": "int", "content": "string", "authorId": "int", "createdAt": "date" }]	✅	**200**	Récupérer les commentaires d'une publication
"DELETE"	"/api/posts/comments/ :id "		{ "status": "success", "message": "Comment deleted" }	✅	**200**	Supprimer un commentaire d'une publication
"PUT"	"/api/posts/comments/ :id "	{ "content": "string" }	{ "status": "success", "message": "Comment updated" }	✅	**200**	Modifier un commentaire par son id
"GET"	"/api/programs"		[{ "id": "int", "name": "string", "description": "string", "createdAt": "date" }]	✅	**200**	Récupérer tous les programmes
"POST"	"/api/programs"	{ "name": "string", "description": "string" }	{ "status": "success", "message": "Program created" }	✅	**201**	Créer un nouveau programme

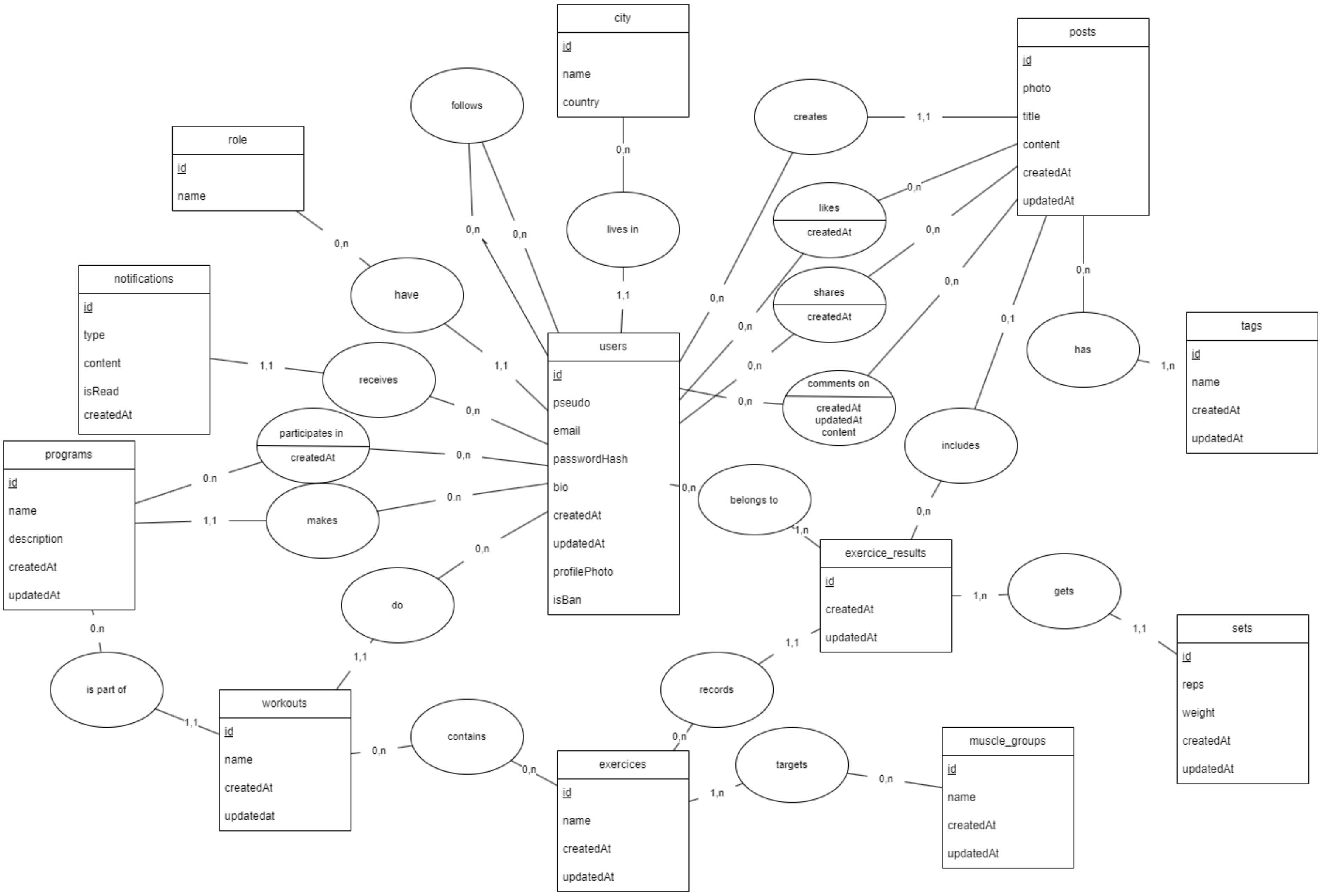
"PUT"	"/api/programs/ :id"	{ "name": "string", "description": "string" }	{ "status": "success", "message": "Program updated" }	✅	**200**	Mettre à jour un programme par son id
"DELETE"	"/api/programs/ :id"		{ "status": "success", "message": "Program deleted" }	✅	**204**	Supprimer un programme par son id
"GET"	"/api/programs/users/:id"		[{ "id": "int", "name": "string", "description": "string", "createdAt": "date" }]	✅	**200**	Récupérer tous les programmes d’un utilisateur par son id
"GET"	"/api/programs/ :id/workouts"		[{ "id": "int", "name": "string", "description": "string", "date": "date" }]	✅	**200**	Récupérer toutes les séances contenues dans un programme par l’id du programme
"POST"	“api/workouts ”	{ "name": "string", "description": "string", "date": "date", "programId": "int" }	{ "status": "success", "message": "Workout created" }	✅	**201**	Créer une nouvelle séance
"PUT"	“api/workouts/ :id”	{ "name": "string", "description": "string", "date": "date" }	{ "status": "success", "message": "Workout updated" }	✅	**200**	Mettre à jour une séance par son id

"DELETE"	“api/workouts/ :id”		{ "status": "success", "message": "Workout deleted" }	✅	**204**	Supprimer une séance par son id
"GET"	“api/workouts/ :id/exercices”		[{ "id": "int", "name": "string", "description": "string", "muscleGroups": ["string"] }]	✅	**200**	Récupérer tous les exercices d’une séance par l’id de la séance
“GET”	“api/tags”		[{ "id": "int", "name": "string" }]	✅	**200**	Récupérer tous les tags
“POST”	“api/tags”	{ "name": "string" }	{ "status": "success", "message": "Tag created" }	✅	**201**	Créer un nouveau tag
“PUT”	“api/tags/ :id”	{ "name": "string" }	{ "status": "success", "message": "Tag updated" }	✅	**200**	Mettre à jour un tag par son id
“DELETE”	“api/tags/ :id”		{ "status": "success", "message": "Tag deleted" }	✅	**204**	Supprimer un tag par son id
“GET”	“api/users/me”		{ "id": "int", "username": "string", "email": "string" }	✅	**200**	Récupérer les informations de l’utilisateur connecté
“PUT”	“api/users/ :id”	{ "pseudo": "string", "email": "string", "bio": "string", "profilePhoto":	{ "status": "success", "message":	✅	**200**	Modifier un utilisateur par son id

		"string", "isBan": "boolean" }	"User updated" }			
"GET"	"/api/users/suggested"		[{ "id": "int", "username": "string" }]	✅	**200**	Récupérer les utilisateurs suggérés pour un utilisateur
“GET”	“/api/users/:id		{ "id": "int", "username": "string", "email": "string" }	✅	**200**	Récupérer les informations d’un utilisateur par son id
"GET"	"/api/users"		[{ "id": "int", "username": "string", "email": "string" }]	✅	**200**	Récupérer tous les utilisateurs
"POST"	"/api/users/follow"	{ "userId": "int" }	{ "status": "success", "message": "Followed user" }	✅	**201**	Suivre un utilisateur
"DELETE"	"/api/users/unfollow"	{ "userId": "int" }	{ "status": "success", "message": "Unfollowed user" }	✅	**204**	Ne plus suivre un utilisateur
"GET"	"/api/notifications "		[{ "id": "int", "message": "string", "read": "boolean" }]	✅	**200**	Récupérer toutes les notifications

"GET"	"/api/notifications/users/ : id "		[{ "id": "int", "message": "string", "read": "boolean" }]	✅	**200**	Récupérer toutes les notifications d'un utilisateur
"POST"	"/api/notifications "	{ "message": "string", "userId": "int" }	{ "status": "success", "message": "Notification created" }	✅	**201**	Créer une notification pour un utilisateur
"DELETE"	"/api/notifications/ :id"		{ "status": "success", "message": "Notification deleted" }	✅	**200**	Supprimer une notification
"PUT"	"/api/notifications/ :id"	{ "type": "string", "content": "string", "isRead": "boolean" }	{ "status": "success", "message": "Notification updated" }	✅	**200**	Mettre à jour une notification par son id

Modèle conceptuel des données



Modèle logique des données

exercices = (id, name, created_at, updated_at);

exercice_results = (id, created_at, updated_at, #exercices_id);

muscle_groups = (id, name, created_at, updated_at);

city = (id, name, country);

role = (id, name);

sets = (id, reps, weight, createdAt, updatedAt, #exercice_results_id);

tags = (id, name, createdAt, updatedAt);

users = (id, pseudo, email, password_hash, bio, created_at, updated_at, profilePhoto, isBan, #role_id, #city_id);

posts = (id, photo, title, content, created_at, updated_at, #author_id, #exercice_result_id);

programs = (id, name, description, created_at, updated_at, #workouts_id);

workouts = (id, name, created_at, updated_at, #programs, #author_id);

notifications = (id, type, content, isRead, createdAt, #users_id);

shares = (#users id, #posts id);

likes = (#users_id, #posts id_, liked_at);

comments = (#users_id, #posts id_, content, commented_at, updated_at);

users_programs = (#users id, #programs id, createdAt);

exercices_workouts = (#exercices id, #workouts id);

exercices_muscle_groups = (#exercices id, #muscle_groups_id);

follows = (#follower id, #followed id);

exercices_results_users = (#exercices_resultsid, #users id);

posts_tags = (#posts_id, #tags id);